A Traffic Partition Algorithm for Switched LANs and Its Performance Analysis

Ding Wei , Gong Jian & Yu Xiao
Southeast University, Nanjing 210096

ABSTRACT

An algorithm is proposed which can be used for the topology design of switched LAN with heavy traffic and multi-segments. The main principle of the algorithm is to split the whole traffic to each segment as average as possible. The algorithm consists of binary division and ordinary division. When the number of segments to be divided equals to powers of 2, binary division is used; ordinary division is based on binary division but suitable to more common cases. Both correctness and time complexity of the algorithm are discussed in detail, and a comparison of the algorithm with the best result is given out at the same time.

Key words: Computer network, Network Design, Algorithm, Switched LAN

1. Introduction

SIN-Switched Internetwork is the main structure of various enterprise network nowadays. The core equipment of those networks is the SWITCH. Usually, the Switch has a high capacity of transmission which makes the various segments attached can share a bandwidth above 10M, so as to satisfy various application requirements which are based on the client/sever schema. The overall Star Structure of SIN makes it easier to more advanced techniques such as ATM, etc.

Even though the bandwidth of Switch's internal bus comes up to several Gs, the partition of ports is still a very important factor that influences the performance of networks in the design of networks with Switches. It is because any information pass through the Switch has to be transmitted through the Switch has to be transmitted through two Ethernets with 10M each. An ideal partition method is that one client/sever group is sited in one port. But in practical situation, it is more common that each client is the client of various sever, while one sever may be a sever of another sever. Under such complex circumstances, the question is how to distribute the traffic of the various ports to make a reasonable structure. Theoretically speaking, it is a matter of topology design. But the conventional algorithm is not suitable for SIN. This paper will give out an algorithm for partition of network segments. The main principle of the algorithm is to split the whole traffic to each segment as average as possible. It consists of binary division and ordinary division. When the number of segments to be divided equals to powers of 2, binary division is used; ordinary division is suitable to more common cases.

2. Definitions and Assumptions

Before discussing and analyzing the algorithm, some definitions and assumptions are described below.

Assumption 1: For the delay of transmitting information in Switched Ethernet: di, when the bandwidth is a constant, di is dependent on and only on the traffic of the subnet: xi, i.e. di=f(xi), and f is a monotone increasing function.

Definition 1: Assume that there are n ports in a Switched Ethernet, the delay of the n ports is d1,d2,...,dn; then the delay of this Switched Ethernet is D=di=max(d1,d2,...,dn), and i is called the maximum delay subnet.

Definition 2: Assume that there are n ports in a Switched Ethernet, the traffic

of each subnet connected to each port is x1,x2,..,xn; If xj=Max(x1,x2,..,xn), then j is called a subnet with the maximum traffic. Let X=xj, where X is the maximum traffic.

Definition 3: Under certain circumstances, assume that there are k methods of traffic partition, whose delay is D1,D2,...,Dk; D1=Min{D1,D2,...,Dk}, then 1 is called the optimal partition.

According to definition 3, the procedure of traffic partition is actually the one of finding the optimal partition 1 as defined above.

Conclusion 1: If the assumption 1 holds true, then the maximum delay subnet discussed in definition 1 is the same as the subnet with the maximum traffic in definition 2.

Proof to this conclusion is omitted. The significance of this conclusion lies in turning the delaying condition of finding the optimal partition to traffic condition. What's more, the latter is easier to compute.

Deduction: The optimal partition in definition 3 make X1=Min(X1,X2,...,Xk) hold true at the same time.

3. Binary division

Since the number of ports in a Switch usually comes to the powers of 2, such as 4, 8, 16, 32, etc., for these Switches the traffic can be partitioned binaryly.

Under any circumstances, when dividing a subset into two sets:

a & b, the internal traffic of the original subset is divided to three parts: the internal traffic of subset a: Xa'; the internal traffic of subset b: Xb'; the traffic between a & b: Xab. Since Xab actually passes through the two ports of the Switch, the actually traffic in the a subset is:

$$Xa=Xa'+Xab \qquad (1)$$

for the same reason:

$$Xb=Xb'+Xab \qquad (2)$$

When binary-dividing S into a & b, if a is a subset of S, the b is then the surplus set. According to the Power-Set theorem in the set theory, the number of subsets of set S with a capacity of n is 2.

In consideration of the symmetry of a & b and their impossibility of being empty set or full set, the ways to binary-divide the S above comes up to 2-1. According to definition 3, now k=2-1. Just to try all these 2-1 methods, the optimal partition 1 will be found.

But when n=50, 2-1=5.6295*10. It may take a workstation with a speed of 10MIPS to execute 5630 days to find out the result, and this is impractical.

To improve the computation efficiency, a method of set binary-division is proposed. It may reach the optimal partition in a certain condition within a fairly good time complexity. The main idea of this method is to transfer the elements of b to set a gradually. Thus the traffic in b will decrease while a increase will occur in set a. At last there will be a balance between a & b, and the algorithm comes to an end. According to the principle that it is better to make related information to flow within one net, the selected element in b always has the maximum traffic connection with set a. The binary-division method

is described below in detail:

(1) Let a=0, b=S, Xa=0, Xb=Xs, the subset with the maximum traffic c=b, the maximum traffic Xc=Xb;
(2) Select one element from b to a, thus |a|=1, |b|=n-1;
(3) Compute Xa and Xb according formula (1) & (2), and find the subset c' with the maximum traffic according definition 2;
(4) If Xc'>=Xc, then there is no improvement and the modification should not be recorded. Otherwise record the modification this time and select the element in b which has the maximum traffic connection with set a and add it to a. Goto (3).

The algorithm can be formally described below :

```
Procedure binary-division (n:int, F:array[1..n, 1..n] of real); /* divide n
hosts to two subsets, where F is the matrix of traffic */
var a,b: set of 1..n; /* used to store the result of division */
     fab: array[1..n] of int; /* record the traffic of each element in b with a
*/
     xa,xb: real; /* refer to formula (1) & (2) */
Begin /* initialize the variables */
  repeat  t= put -atob(fab,a,b,xa,xb)
  until   (not t);
/* put-atob is a boolean function, it selects the element with the maximum
traffic in b and add it to a. If there is a improvement then ,a new division
take the place of the old one, and return 'True', or return 'False'. */
   output the result of division of a & b;
End;
```

Put-atob is described below:

```
function put-atob(var fab:array[1..n] of real; var xa,xb:real; var a,b:set of
1..n);
  begin /* initialize the variables */
     put-atob=true;
    find the element k in b which has the maximum traffic with a;
    pre-modify xb;
    pre-modify xa;
    if      there is any improvement,
    then a new division takes the place of the old one.
    else  put-atob=false
 end;
```

The correctness and time complexity of the above algorithm can be proved by the following theorems.

Theorem 1: If there is fab[k]>0 in matrix fab (i.e. the traffic between a & b doesn't equal to zero), and put-atob is true, then the traffic in b is monotone decreasing. That is, let xb0 become the traffic before executing, xb1 become the traffic after executing, the xb1<xb0.

Proof: Assume before modification b={b1,b2,...,bk,...,bm}, put bk to a after modification. Diagram 1 & Diagram 2 describe the changes of traffic before and after modification. The identifiers are the same as defined in formula(1) & (2).

According to formula (2) & Diagram 1, Diagram 2:

Theorem 2: If n (the no. of hosts)> 3, and at least two pairs of hosts have traffic which is greater than zero, then the function put-atob will not make b empty.

The proof is omitted. Its significance lies in the proof that put-atob will terminate.

Theorem 3: The time complexity of algorithm put-atob is o(|b|).

Theorem 4: The time complexity of algorithm binary-division is o(|n|).

The proof to the above two theorems is omitted.

Every step of the algorithm is dependent on the state of partition at present. Thus it may not reach the optimal partition as enumeration does. In the algorithm above, the first step of partition (i.e. set a is empty, select a element from b to a) is the basis of the whole operation. Thus the first step is of significance. But the selection in the algorithm is random. If another circle is added to the binary-division to begin the above algorithm with each element in turn (it can be done by the initialization in a), then it can be in a wider range to find the optimal result. At this time, the time complexity of the algorithm goes up from nto n. But it is still within the limit of tolerance.

4. Ordinary Division

The binary division can only divided the set of nodes into two. Even though it may satisfy the requirements in most cases, by dividing continuously to get subsets with the amount of the powers of 2, there are some Switches that have ports with the amount which can not be divided by 2, such as 6, 12, etc.. Thus it is worth to discuss how to divide the set of nodes to random number of subsets based on the binary division.

Since it is more complex for ordinary division than binary division, it is still impossible to find the optimal one with enumeration. Assume m is the number of subsets to be divided. The main idea of ordinary division is: Select the subset with the maximum and minimum traffic within n sets, modify and allocate the traffic to balance.

This algorithm can be done in two ways, and discussed in detail below.

Directly division: Set up m sets from the very beginning, among which one is a full set, the other n-1 sets are empty. Go on dividing as above until arriving at balance.

Iterative division: It starts with binary division . Add one set after each balance until the number of sets reaches m.

Algorithm of direct division:

Step 1 S1={1,2,...,n}; S2=S3=...=Sm={};
Step 2 find the subset with the maximum traffic Smax and the subset with the minimum traffic Smin from S1 to Sn.

Step 3 add one node in Smax with the maximum traffic with Smin to Smin
Step 4 goto step 3 until balance is reached .
Step 5 goto step 2, step 3, and step 4 until balances are reached.

Algorithm of iterative division:

Step 1 S1={1,2,...,n}; k=2; /* k is the control variable of iteration*/
Step 2 Sk={};
Step 3 find the subset with the maximum traffic Smax and the subset with the minimum traffic Smin from S1 to Sk;
Step 4 add one node in Smax with the maximum traffic with Smin to Smin .
Step 5 goto step 4 until balance is reached;
Step 6 k++ ; if k>m then end, otherwise goto step 2.

In the two algorithms above , the flow mode of node and the principle of balance is the same as binary division. But it should be noticed that the flow of nodes will not only change the traffic between the source set and the object set, it also effect the traffic among these two sets and the other sets. The traffic related to source set should be added to the object set. However, it is obvious that the actual traffic (i.e. the traffic within and between the sets) of all the other sets will not change except the two sets related to flow of nodes.

The two algorithms have been realized on the SUN workstation using C. Through comparing various simulation data, it has been found out that there is no obvious differences in the comparatively important segments partition (6-12).

5. The Comparison of Ordinary with the Best Result

We compared the ordinary division (iterative division) with the best result obtained by numeric method under the 14 groups of Monte-Carol simulation traffic (with the normal distribution, exponential distribution, equidistribution and logarithmic distribution) when n=10 and m=3. The result is shown bellow(table 1):

Original Traffic amount	difference of max traffic	relative difference(%)	difference of average traffic
27.14	1.03	3.80	0.01
33.81	1.25	3.70	-0.69
46.64	3.15	6.75	

0.44

77.59
4.24
5.46
-0.20

87.26
7.83
8.97
1.00

93.90
1.00
1.06
-0.39

102.52
8.75
8.53
-0.72

115.50
9.54
8.26
0.70

117.93
0.37
0.31
-2.88

118.89
0.15
0.13
0.24

139.74
2.64
1.89
0.10

190.12
2.32
1.22
0.86

252.45
2.60
1.03
-2.28

298.80
8.58
2.87
-1.44

Table 1 comparison of iterative division with the best result
within the table:
difference of max traffic = max traffic of iterative division - max traffic of numeric division (the best result)
relative difference = difference of max traffic / original traffic amount
difference of average traffic = average traffic of sets in iterative division - every traffic of set in numeric division

As the target, the termination condition, of the both methods is to minimize the max traffic (see definition 2) parameter of the division, the difference of average traffic may be negative.

In the table, the biggest gap between iterative division and the best result is 8.97%, the smallest one is only 0.13%, and the average is 4.20%. The average gap of the first seven groups in the table is 5.46%, and the last seven groups is 2.24%. This means the iterative division is more suitable to the largescale network in which the traffic is more heavy. On the other hand, it is little obvious difference on the average traffic parameter between the two methods, and the iterative division is even a little better.

About the time complexity analysis, the numeric method is 3(m), and that of the iterative division is only 3*10(m*n), that shows the latter one is much better than the former, and the defference will become more and more abvious as m and n increase.

6. Conclusions

With the wide use of computers, the density of microcomputers can be very high in a small area in most cases. When such new equipment as Switch are used in the design of Switched LANs to support the client/server schema and the new techniques such as multimedia applications, the problems that has to be faced up are basically different from the design of wide area network before. The Switched network are now taking the place of conventional shared-media network gradually and become an important part of the technique of enterprise network. But at present there are few practical network design methods adapted to this new situation. This problem has aroused more attention . Related research is becoming hot spot.

At the same time, various techniques such as simulation, system engineering and artificial intelligence etc. have been used to solve the problems of this research field. This paper is based on the principle of traffic balance and gives out a partition algorithm. It is suitable to the situation above with a good practicality. This algorithm can also be used in the topology design for wide area networks to group the nodes into hierarchical structure during the initial phase of the design.

7.References

(1) Robert Mandeville. Ethernet Switched Evaluated. DATA COMMUNICATION, McGRAW-HILL'S NETWORKING TECHNOLOGY MAGAZINE. MARCH 1994
(2) Bradley F. Shimmin. Comparing Three Methods of Ethernet Switching. LAN TIMES, McGRAW-HILL'S INFORMATION SOURCE FOR NETWORK COMPUTING, VOL.11 ISSUE. 1 1994
(3) K.Kuratowski. & A.Mostowski. Set Theory. North-Holland New. York, 1976
(4) L.J.Leblane and S.Narasimhan. Topological expansion of metropolitan area

network. Computer Network and ISDN System, Vol.26, No.9 1994 pp1235-1284
(5) H.Saito and T.Asaka. Traffic aspect of personal telecommunications in intelligent network. Computer Networks and ISDN Systems , Vol. 26, No. 9 pp 1089-1100

The paper's work is supportted by the Jiangsu provincial key laboratory of Computer Network Technology.

交换式互连局域网的流量分割算法及性能分析

丁伟　　龚俭　　余晓
南京　210096　　东南大学计算机科学与工程系

摘　要

本文提出了一个适用于大流量、多网段环境的交换式互连局域的网段划分算法，其基本原则是流量均衡。算法具体又分为二分划分和普通划分两部分。二分划分适用于网段数为二的整数次幂的划分；普通划分的原理基于二分划分，但适用于更一般的情况。文中对算法的正确性和时间复杂性等问题进行了比较详细的分析和讨论，同时给出了该算法与最佳解的比较。

关键词：计算机网络；网络设计；交换式局域网；算法